

LISTING OF CLAIMS

1-8. (Cancelled).

9. (Currently Amended) A method, comprising:

receiving a request to store a first firmware variable;

determining if a compressor **stored in a non-fault tolerant portion of a firmware storage device** is available for compressing the first firmware variable; and

employing the compressor if it is available to compress the first firmware variable and ~~store it~~ **storing the first firmware variable in a compressed form** in a firmware storage device, **the first firmware variable in the compressed form to be decompressed via a decompressor stored in a fault-tolerant portion of the firmware storage device,** otherwise storing the first firmware variable in an uncompressed form in the firmware storage device if the compressor is not available.

10. (Original) The method of claim 9, further comprising:

receiving a request to store a second firmware variable;

determining the compressor is no longer available; and

storing the second firmware variable in the firmware storage device in an uncompressed form.

11. (Original) The method of claim 9, wherein uncompressed firmware variables are stored in a 2-tuple format of

$\langle M_i, B_i \rangle$,

wherein M_i comprises metadata corresponding to an i th tuple, and B_i comprises data corresponding to the i th tuple, while compressed firmware variables are stored in a 2-tuple format of

$$\langle M'_i, C(B_i) \rangle,$$

wherein M'_i comprises metadata corresponding to an i th tuple containing indicia indicating the i th tuple is compressed, B_i comprises data corresponding to the i th tuple, and C represents a compression function.

12. (Original) A method, comprising:
- in response to a computer system power-on or reset event,
 - scanning a firmware storage device in the computer system for uncompressed firmware variables that are stored in an uncompressed form;
 - converting the uncompressed firmware variables to a compressed form.
13. (Original) The method of claim 12, wherein the uncompressed firmware variables are converted to a compressed form by performing operations including:
- copying an image of a firmware memory block in which the uncompressed firmware variables are stored;
 - erasing the firmware memory block;
 - compressing each uncompressed variable; and
 - writing the compressed variables back to the firmware memory block.
14. (Original) The method of claim 13, further comprising:

scanning the image for any compressed firmware variables; and
writing the compressed variables back to the firmware memory block.

15. (Currently Amended) A method comprising:

storing a first converter **and a second converter** in a non-fault tolerant portion of a
firmware storage device;

storing a first deconverter **and a second deconverter** in a fault tolerant portion of the
firmware storage device;

determining if **each of** the first converter **[[is]] and the second converter are** available;
and

storing firmware variables in a **combined converted form via first and second
conversion operations performed by the first and second converters respectively if it is
determined that both of the first and second converters are available;**

**storing firmware variables in a first converted form via conversion operations
performed by the first converter** if the first converter is determined to be available **and the
second converter is not available;**

**storing firmware variables in a second converted form via conversion operations
performed by the second converter if it is determined that the second converter is available
while the first converter is not available;**

otherwise storing the firmware variables in an unconverted form.

16-21. (Cancelled).

22. (Currently Amended) An article of manufacture, comprising:
a machine-readable **non-transitory storage** medium on which instructions are stored,
which when executed facilitate storage of firmware variables by performing operations
including:
receiving a request to store a first firmware variable;
determining if a compressor **stored in a non-fault tolerant portion of a
firmware storage device** is available for compressing the first firmware variable; and
employing the compressor if it is available to compress the first firmware variable
and ~~store it~~ **storing the first firmware variable in a compressed form** in a firmware
storage device, **the first firmware variable in the compressed form to be
decompressed via a decompressor stored in a fault-tolerant portion of the firmware
storage device,** otherwise storing the first firmware variable in an uncompressed form in
the firmware storage device if the compressor is not available.
23. (Original) The article of manufacture of claim 22, wherein the article comprises flash
memory.
24. (Currently Amended) The article of manufacture of claim 23, wherein ~~a portion of the
instructions comprise a compressor employed for compressing firmware variables and~~ the flash
memory includes ~~[[a]]~~ **the** non-fault tolerant block of memory in which the compressor is stored.

25. (Original) The article of manufacture of claim 24, wherein execution of the instructions performs the further operations of:

receiving a request to store a second firmware variable;

determining the compressor is no longer available; and

storing the second firmware variable in the firmware storage device in an uncompressed form.

26. (Currently Amended) The article of manufacture of claim 23, wherein ~~a portion of the instructions comprise a decompressor employed for decompressing compressed firmware variables and~~ the flash memory includes ~~[[a]]~~ the fault-tolerant block of memory in which the decompressor is stored.

27. (Currently Amended) A computer system, comprising:

a motherboard;

a processor, coupled to the motherboard;

volatile memory, coupled to the motherboard; and

a boot firmware device, coupled to the motherboard and comprising flash memory having firmware components stored therein including a compressor, the firmware components comprising instructions that when executed by the processor effectuate storage of firmware variables by performing operations including:

publishing an interface;

receiving a request to store a firmware variable via the interface;

determining if a compressor **stored in a non-fault tolerant portion of a firmware storage device** is available for compressing the firmware variable; and
employing the compressor if it is available to compress the first firmware variable and store it **storing the firmware variable in a compressed form** in a firmware storage device, **the firmware variable in the compressed form to be decompressed via a decompressor stored in a fault-tolerant portion of the firmware storage device,**
otherwise storing the firmware variable in an uncompressed form in the firmware storage device if the compressor is not available.

28-30. (Cancelled).

31. (New) The method of claim 15, wherein the first converter comprises a compressor, and the first deconverter comprises a decompressor.

32. (New) The method of claim 31, wherein the second converter comprises an encryptor and the second deconverter comprises a decryptor.

33. (New) The method of claim 32, wherein the second converter adds error correction code (ECC) data to a variable, and the second deconverter utilizes the ECC data detect and correct bit errors in non-ECC data of the variable.